INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume2, Issue6)

Available online at www.ijarnd.com

A Pairwise Algorithm using Speech Detachment and pitch Estimation for The Deep Stacking Network

S. Saranya, E. Menaka

[1]PG Student,[2]Assistant Professor, Department of Computer Science & Engineering Vivekanandha Institute of Engineering & Technology for Women, Thiruchangode, Namakal, Tamilnadu, India

saransarsen@gmail.com

ABSTRACT

The Speech detachment and contribute estimation loud conditions are thought to be a "chicken-and-egg" issue. Pitch data is a one of the essential prompts for discourse partition. Also, discourse detachment makes pitch estimation simpler when foundation commotion is expelled. In this paper, we propose a directed learning design to take care of these two issues iteratively. The proposed calculation depends on the profound stacking system (DSN), which gives a technique to stacking straightforward preparing modules to manufacture profound structures. Every module is a classifier whose objective is the perfect paired veil (IBM), and the information vector incorporates otherworldly components, pitch based elements and the yield from the past module. Amid the testing stage, we gauge the pitch utilizing the partition results and refresh the pitch-based components to the following module. At the point when inserted into the DSN, pitch estimation and discourse detachment each run a few times. To examination, technique demonstrates that the proposed framework result in both a top notch assessed twofold cover and precise pitch estimation and outflanks late frameworks in its speculation capacity.

Keywords: Speech Separation, Pitch Estimation, Computational Auditory Scene Analysis, Supervised Learning, Ideal Binary Masking, Deep Stacking Network.

1 INTRODUCTION

Insensible conditions, commotion significantly corrupts the discourse comprehensibility of hearing -disabled audience members and the execution of programmed discourse acknowledgment (ASR) frameworks. Discourse detachment plans to evacuate clamor by isolating target discourse from foundation impedance. It is important for both amplifier plan and ASR frameworks [2], [3]. Different methodologies have been proposed for discourse partition including ghostly subtraction [4], free segment investigation and model-based approaches. Computational sound-related scene examination (CASA) is a promising **technique that endeavors to** copy the human sound-related framework and isolates a sound blend in light of perceptual standards.

1.1Computational Auditory Scene Analysis

CASA defines one of the speech separation goals as computing an ideal binary mask (IBM). The IBM is a time-frequency (T-F) mask that can be computed from a premixed target and interference. Specifically, for each T -F unit, if the signal-to-noise ratio (SNR) is greater than a local SNR criterion, the corresponding mask element in the IBM is set to 1 (target dominant). Otherwise, the mask element is set to 0 (interference dominant). Adopting IBM as the computational goal of CASA, a series of studies have shown that processing noisy speech using g the IBM

can substantially improve speech intelligibility for humans. Recent experiments also show that speech or speaker recognition performance can be greatly improved by using IBM.

1.2 Ideal Binary Masking

To estimate the IBM, separation algorithms must rely on the intrinsic characteristics of the corrupted input signal. A standard CASA-based separation system always decomposes the input signal into time -frequency units. Then, it uses some cues to group the T-F units into different sources. Recently, IBM estimation has been treated as a binary classification problem and solved by supervised algorithms such as the deep neural network (DNN). The generalization problem is the biggest issue with supervised methods. One way to solve the generalization problem is to expand the training set to cover more application scenarios. Another way is to use more robust features with invariant properties. For speech separation, the pitch is one such feature; however, pitch estimation in noisy conditions is also a challenging task, especially when the SNR is low. Many studies have been done with regard to robust pitch estimation. In fact, speech separation and pitch estimation were also considered to be a "chicken -and-egg" problem [10]. On one hand, pitch information can dramatically improve the performance of a speech separation system. On the other hand, pitch estimation becomes quite easy when the speech is separated from the background noise.

1.3 Deep Stacking Network

The deep stacking network (DSN) is to combine speech separation and pitch estimation [11]. These two problems boost each other iteratively. DSN stacks simple processing modules to build forward deep architectures by feeding the outputs of each lower module into the next higher one. The basic module is a shallow neural network, whose job is to estimate the same target. For our work, the target is IBM and the input is frame -level features consisting of spectral and pitch-based features. DSN provides a flexible way to add additional processes at each "Stacking" step. In this study, after obtaining the estimated binary mask (EBM) from a basic module, we use the EBM to re-estimate the pitch and then update the pitch-based feature. The EBM and the updated features are provided to the next basic module as input. Finally, the EBM obtained from the last module is regarded as the separation result. The pitch estimation is also obtained based on the final EBM. The main contribution of this work is that we insert the traditional signal processing module, pitch estimation, into the DSN which is a supervised learning algorithm with deep architecture for speech separation.

2 RELATED WORKS

H. Zhang [1], Examined pitch data is an essential sign for discourse partition. Since contribute estimation uproarious condition is a testing task. However, the proposed a review administered learning design which consolidates these two issues succinctly. The proposed calculation depends on profound stacking system (DSN) which gives a technique for stacking basic handling modules in building profound.

Y. Shao [2], Suggest a conventional automatic speech recognizer does not perform well in the presence of multiple sound sources, while human listeners are able to segregate and recognize a signal of interest through auditory scene analysis. To present a computational auditory scene analysis system for separating and recognizing target speech in the presence of competing for speech or noise.

S. Boll [3], Spectral subtraction has been shown to be an effective approach for reducing ambient acoustic noise in order to improve the intelligibility and quality of digitally compressed speech. They will present a set of implementation specifications to improve algorithm performance and minimize algorithm computation and memory requirements. It is shown spectral subtraction can be implemented in terms of a no stationary, multiplicative, frequency domain filter which changes with the time-varying spectral characteristics of the speech.

3 SYSTEM ARCHITECTURE

The discourse partition and pitch estimation are considered as a "chicken -and-egg" issue. To address this issue, we prepare discourse detachment and pitch estimation on the other hand. The thought is to first acquire an unpleasant gauge of the IBM, and after that utilization, it to assess the objective pitch. From the assessed pitch, we can create a superior isolation which can in this manner be utilized for a superior pitch estimation. The guide appears in the calculation Estimate IBM and pitch (Fig. 1). In the calculation "Appraise IBM and pitch," the information is a blend (loud discourse), and the yields are the EBM and the assessed pitch. The calculation includes three procedures: "Gallus" utilizes the blend to appraise the IBM; "Egg" utilizes the blend and the EBM to gauge pitch; lastly "Chicken" uses the blend and the assessed pitch to evaluate the IBM. The last two procedures run then again, we allude them together as "Chicken-and-Egg," and they, on the other hand, support the precision of the pitch

estimation and discourse detachment assignments. We developed the framework shown in Fig. 2 The proposed framework including two phases, one is a preparation stage and second is a trying stage. In the preparation organize, premixed discourse and clamor are used to build the IBM, which is the preparation target. Here, we utilize outline level elements as info. The elements are separated from the blended flag, which comprises of both ghostly elements and pitch-based components. To figure the pitch-based components, we utilize ground truth contribute the preparation organize, while in the testing stage, the pitch utilized is assessed iteratively by the proposed strategy. The features used in this work consist of spectral features and pitch-based features extracted from noisy speech.

Algorithm Estimate IBM and pitch
Input: Mixture
Output: EBM, Pitch

EBM ← Gallus(Mixture)
for $i = 1$ to N **do**
  Pitch ← Egg(Mixture, EBM)
  EBM ← Chicken(Mixture, Pitch)
end for

Fig. 1. Algorithm Estimate IBM and pitch, where N is the number of iterations. This idea is inspired by the famous "Chicken or the egg, which came first?" dilemma. The answer from the evolutionary theory is that a creature named "Gallus"-similar to a chicken but not a chicken-laid the first chicken eggs. These eggs then hatched into chickens that inbred to produce a living chicken population. (For more information, see

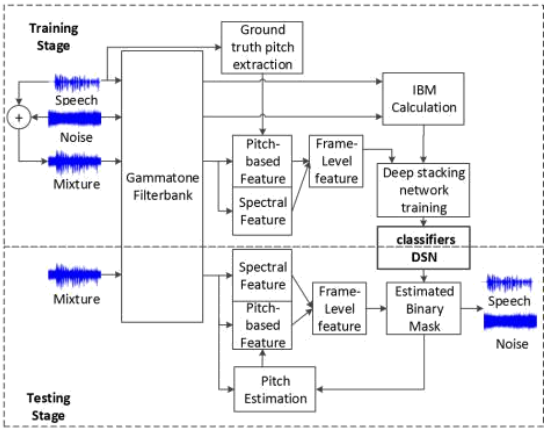

Fig 2: The architecture of proposed layer

a) Spectral Features

After decomposing the signal into T-F representation, we compute the energy of each T-F unit (*c*,*m*) by summing the square of the filter responses in it and then compresses the energy by using the following cubic root operation:

$$E(c,m) = \sqrt[3]{\sum_n g^2(c, mT - n)} \quad (1)$$

Where refers to the frequency channel, *m* refers to the time frame, *g* is the filter response, $T = 160$ corresponds to a 10 ms frame shift, and $n \in [1,160]$. This energy matrix is called a cochleagram.

- Pitch-based Features

Our pitch-based features are derived from normalized autocorrelation functions (ACF) and envelope ACF. ACF (*A*) and envelope ACF (An_E) are computed as shown below:

$$A(c,m,\tau) = \frac{\sum_n g(c, mT - n)g(c, mT - n - \tau)}{\sqrt{\sum_n g(c, mT - n)^2 \sum_n g(c, mT - n - \tau)^2}} \quad (2)$$

$$A_E(c,m,\tau) = \frac{\sum_n e(c, mT - n)e(c, mT - n - \tau)}{\sqrt{\sum_n e(c, mT - n)^2 \sum_n e(c, mT - n - \tau)^2}} \quad (3)$$

Where is the envelope of *g*, which is obtained by a Hilbert transformation. Given a modulated waveform, the envelope can be constructed from the absolute value of the analytic signal. This analytic signal is complex and is composed of the original waveform as the real part and its Hilbert transform as the imaginary part. More detail concerning the analytic signal and the Hilbert transform can be found in the delay, $\tau \in [0, 12.5]$ ms, corresponds to the pitch period, and the maximum corresponds to an 80 Hz pitch. The ACF is called a correlogram and is widely used for pitch estimation and source separation. The envelope ACF depicts the amplitude modulation rate in the high-frequency channels.

- Frame-level Features

To estimate the IBM, we train a classifier using the frame level features, which are formed by simply combining the unit level features of all the channels in a frame. The frame-level features include spectral features (*E*) and pitch-based features (*A*, A_E).

$$F_{unit}(c,m) = \begin{pmatrix} E(c,m) \\ A(c,m,\tau_m) \\ A_E(c,m,\tau_m) \end{pmatrix} \quad (4)$$

$$F_{frame}(m) = \begin{pmatrix} F_{unit}(1,m) \\ F_{unit}(2,m) \\ \vdots \\ F_{unit}(N,m) \end{pmatrix} \quad (5)$$

In this study, N = 64, and the frame-level features are 192dimensional with 64 dimensions of spectral features and 128 dimensions of pitch-based features.

4 PROPOSED SYSTEM

4.1 IBM Estimation

The IBM Estimation is used from the viewpoint of classification, we estimate the IBM by supervised trained classifiers. As described in the previous section, we use frame -level features for IBM estimation. The classifiers of all channels share the same features and predict IBM frame by frame. This is different from the procedure used in [21], which employed different features to train a sub-band classifier. For training, we set the input $X = F_{rame}(m)$ and the training target $T = IBM(m)$. Here, $IBM(m)$ is the vector of IBM at frame m:

$$IBM(m) = \begin{pmatrix} ibm(1,m) \\ ibm(2,m) \\ \vdots \\ ibm(N,m) \end{pmatrix} \quad (6)$$

We train the classifiers by maximizing the hits minus the false alarm rates (HIT -FA) [16], which is a widely used evaluation metric and has been shown to be highly correlated with human speech intelligibility. H IT-FA is calculated as *HIT* minus *FA*,

$$HIT = \frac{\#\text{Hit}}{\#\text{Hit} + \#\text{Miss}}, \quad FA = \frac{\#\text{False alarm}}{\#\text{False alarm} + \#\text{Reject}}$$

Where #*x* indicates the number of *x* and *x* is defined as Tab. I. A higher HIT -FA rate indicates a better speech separation performance.

TABLE CALCULATION OF HIT-FA.

	IBM	Estimated IBM
Reject	0	0
False Alarm	0	1
Miss	1	0
Hit	1	1

To make the result more useful for speech separation, the classifiers are trained by maximizing HIT-FA:

$$\underset{W,\phi}{\operatorname{argmax}}(\text{HIT-FA}(z,T)) \quad (7)$$

This problem can be solved approximately by two steps. The first step is to optimize W to the minimum mean square error (MSE) between y and T:

$$\underset{W}{\operatorname{argmin}}(\text{MSE}(y,T)) \quad (8)$$

The second step is to optimize ϕ to maximize the HIT-FA of z with reference to T, where ϕ is used as a binary threshold:

$$\underset{\phi}{\operatorname{argmax}}(\text{HIT-FA}(z,T)) \quad (9)$$

The inputs of this model are the frame -level features, which are formed by combining the unit-level features of all channels in a frame. With different definitions of unit-level features, we could use different information. The "Gallus" process is defined to use only spectral features for the initial IBM estimation, where no pitch information ion is involved the features would include both the spectral and pitch-based features. We use that in the "Chicken" process of our roadmap.

4.2 Pitch Estimation

The differences between the tandem algorithm and the proposed algorithm are followed. First, the tandem algorithm used only the pitch-based features, so it didn't address unvoiced speech. In the proposed algorithm, both spectral and pitch-based features are used; therefore, it is applicable to both voiced and unvoiced speech separation. Second, within each iteration, their algorithm used the same MLPs and, consequently, the entire system was based on a shallow network, which limits its modeling ability. In contrast, the proposed algorithm is based on the DSN which is a deep network that is more powerful than a shallow network.

4.3 Combining Speech Separation and Pitch Estimation

After that the speech separation and pitch estimation method in the previous two subsections, we have all the pieces required for our idea to solve those two problems alternately. In this subsection, we introduce the method to combine these two problems.

In fact, we combine the speech separation and pitch estimation by building a relationship between their inputs and outputs. From speech separation to pitch estimate ion, the "Egg" process, the output of speech separation is the EBM, an $L(c,m)$ for every unit (c,m). With Formula (15), the $L(c,m)$ are is applied to update the summary correlogram, then the summary correlogram is used for pitch estimate ion. From pitch estimate ion to speech separation, the "Chicken" process, the outputs of pitch estimation are the pitch period sequence, an $\tau_m$for every frame m. With Formula (4), the $\tau_m$is applied to update the pitch-based features; then, the pitch-based features are used for speech separation.

We treat the process where "Chicken" follows "Egg" together as a basic module, "Chicken -and-Egg", which handles both speech separation and pitch estimation. Then, we embed the basic module into a deep stacking architecture to improve the performance, as discussed in the next section.

4.4. Embedding the Chicken-and-Egg module into the Deep Stacking Architecture

We embed the "Chicken -and-Egg" module into a deep stacking architecture, which is a variant of the deep stacking network (DSN). A typical DSN was proposed in [11], which includes a variable number of basic modules. The "deep" architecture is constructed by stacking the basic modules one onto another, with the output of one feeding the input to the next. In this way, higher modules can use the results of lower modules to make their own improvements. The DSN provides a method to stack simple processing modules to build deep architectures. When the performance can improve as more layers are stacked into the architecture.

The variant of the DSN used in this work differs from the original one in three aspects. First, we introduce a context window into the DSN. The original DSN is not designed for sequenced data and only feeds the "current" outputs from a lower module to a higher one. However, we take the signal continuity into consideration. We feed not only the "current" outputs but also the outputs from adjacent frames in the context window into the higher module. Specifically, we feed a window of frames of estimated IBM into the higher module to utilize the context

information. The idea of treating the estimated IBM. Compared with treating the acoustic feature as context, this method can reduce the dimensionality of input feature. Second, we add some additional processes at each "stacking" step. The original DSN feeds the outputs from lower module to the higher one directly; however, in this work, we utilize the flexibility of DSN. Before feeding the outputs to the higher module, we use the outputs to obtain the pitch estimation and update the pitch-based feature.

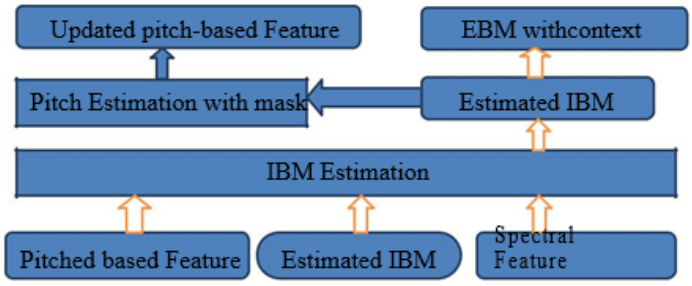

Fig 3 Three layer architecture of DSN

Third, we replace the basic modules with the "Chicken and-Egg" process, as illustrated in Fig. 3. The proposed basic module consists of two processes. 1) The "Chicken" process estimates the IBM using features with pitch information; 2) The "Egg" process refines the pitch estimation by applying the EBM to the summary correlogram and updates the pitch-based features for IBM estimation.

5 COMPARISON METHOD

- Configuration of the Proposed System

The proposed system is stacked with 5 basic modules. The configurations of each module are listed in Tab. II. For the 1st layer, inputs include only the spectral features to estimate IBM. For the 2nd layer, inputs are composed of the output of the 1st layer (estimated IBM) and the spectral features. For the 3rd layer, inputs include the output of 2nd layer, the spectral features, and the pitch-based features. For the 4th and 5th layers, in addition to the spectral and pitch-based features, we also add the context informant ion of the lower layer's outputs into the inputs. The context the window size is 3 and 5 for the 4th and 5th layers, respectively. The network architecture is used in the two types of comparison methods. They are 1. Speech Separation Methods, 2. Pitch Estimation Method.

1. Related Speech Separation Methods: The comparison systems include: GMM -based [35], DNN-based [17] and DNN-SVM -based [17] methods. For the GMM-based method (denoted as "GMM"), we use a 64 –component GMM with diagonal covariance. For the DNN-based method (denoted as "DNN"), we use a DNN with two 200-node hidden layers,

Table 2 CONFIGURATION OF THE PROPOSED SYSTEM.

Layer	Input	Dimension
1	spectral features	64
2	last layer output + spectral features	64+64
3	last layer output + spectral features + pitch-based features	64+64+128
4	last layer output with 3-frame context window + spectral features + pitch-based features	(64×3)+64+128

5	last layer output with 5-frame context window + spectral features + pitch-based features	(64×5)+64+128

which is trained by mini-batch gradient descent with 200 epochs for RBM pertaining and with 100 epochs for network fine-tuning. For the DNN-SVM -based method (denoted as "DNN-SVM"), we combine raw features and the outputs of the last hidden layer in the DNN to train a linear SVM. All three of these methods train a classifier for each channel using unit level features. The unit-level features include 15-D AMS, 13- D RASTA -PLP, 31-D MFCC and 6-D pitch-based features [35]. The features comprise 65-D in total. It should be mentioned that all the speech separation systems are trained using the mixtures at 0 dB because all these systems involve pitch -based features, which are calculated using the ground truth pitch for training and estimated pitch for the test. The estimated pitch is provided by a multi-pitch tracker [36] for these comparison systems.

a) Related Pitch Estimation Methods: We compare our approach with four recently proposed pitch tracking algorithm ms: Jin and Wang's method [36] (denoted as "Jin"), the PEFAC algorithm [37] (denoted as "PEFA C") and Han and Wang's method [38] with DNN and RNN (denoted as "DNN" and "RNN," respectively).

2) Performance of speech separation and pitch estimation

The speech separation method is an easily outperformance the other methods on both noise-matched and unmatched conditions according to the HIT-FA rate. We can also see that our method has only as mall gap between matched and unmatched conditions. This result indicates that the proposed method has a good generalization ability on unseen noises from the result are listed in Table 3.

Table 3 HIT-FA RESULTS OF DIFFERENT METHODS AT 0 DB

	Methods	HIT	FA	HIT-FA
Noise matched	GMM	78.88	31.48	47.40
	DNN	85.64	15.18	70.45
	DNN-SVM	85.29	14.13	71.16
	Proposed	86.39	8.18	78.21
Noice unmatched	GMM	79.02	29.69	49.33
	DNN	87.62	31.77	55.85
	DNN-SVM	87.17	28.93	58.24
	Proposed	89.65	12.92	76.73

To evaluate the generalization to unseen SNRs, we compare the performances of the methods at various SNR conditions. The results are listed in Fig. 4. From Fig. 4, we can see that the proposed system achieves the best generalization for all the SNR-unmatched conditions. It also can be observed that the HIT-FA rates of the proposed system become higher with increasing SNR, while the comparison methods achieve their best results at 0 dB (SNR matched condition) test conditions. The left image shows the performances of the noise-matched condition, and the right image shows the performance of the noise-unmatched condition.

This phenomenon may be because we use spectral subtraction as a preprocessing step, which restricts the input feature space and decreases the distance between matched and unmatched conditions. A further analysis of the effect of spectral subtraction is given in Section VI-F5.

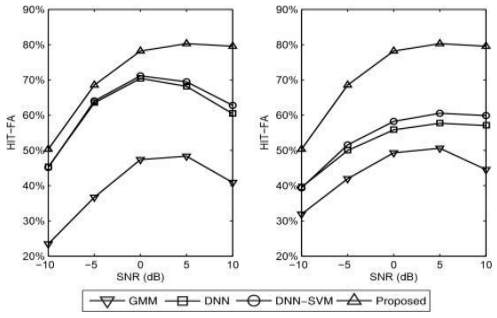

Fig. 4. Overall HIT-FA performances on the different SNR

2) The performance of Pitch Estimation: We compare the proposed system with other four-pitch estimation methods. The proposed method has substantially higher detection rates and lower voicing decision error than the compared approaches. The advantages hold for both noise matched and unmatched conditions, demonstrating that the proposed system generalizes well to new noises.

6 CONCLUSION

We proposed a profound engineering for discourse partition and contribute estimation boisterous conditions that consider these two issues as a "Chicken-and-Egg" issue. We install the "Chicken-and-Egg" handle into a DSN. This approach is more direct than the work in and constitutes a total preparing system. Presently they, profound learning has made awesome progress in many research fields including programme discourse acknowledgment, picture order, and normal dialect preparing. The key considers its prosperity are a lift in computational power and the capacity to deal with a lot of preparing information. The profound neural system (DNN) can catch the shrouded designs in the crude info information the DNN has been considered as a discovery due to its unexplained handling. For discourse partition, the immediate approach to enhance the execution is to incorporate an assortment of loud crude info information. We don't consider that the best approach. In the customary flag preparing field, we have as
of now acquired productive information and standards that are both concrete and effective when their preconditions are fulfilled.

7 REFERENCES

1. H. Zhang, X. Zhang, S. Nie, G. Gao, and W. Liu, "A pairwise algorithm for pitch estimation and speech separation using deep stacking network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2015, pp. 246--250.
2. Yang Shao , Soundararajan Srinivasan , Zhaozhang Jin , DeLiang Wang, A computational auditory scene analysis system for speech segregation and robust speech recognition, Computer Speech and Language, v.24 n.1, p.77-93, January 2010
3. H. Dillon, Hearing Aids. Sydney, Australia: Thieme Boomerang Press, 2001.
4. S. Boll, "A spectral subtraction algorithm for Suppression of acoustic noise in speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'79)*, 1979, vol. 4, pp. 200--203.
5. A. K. Barros , T. Rutkowski , F. Itakura , N. Ohnishi, Estimation of speech embedded in a reverberant and noisy environment by independent component analysis and wavelets, IEEE Transactions on Neural Networks, v.13 n.4, p.888-893, July 2002
6. Alexey Ozerov , Emmanuel Vincent , FrédéricBimbot, A General Flexible Framework for the Handling of Prior Information in Audio Source Separation, IEEE Transactions on Audio, Speech, and Language Processing, v.20 n.4, p.1118-1133, May 2012
7. DeLiang Wang , Guy J. Brown, Computational Auditory Scene Analysis: Principles, Algorithms, and Applications, Wiley - IEEE Press, 2006

